

**Aufgabe 1: (10 Punkte)**

Bei den Multiple-Choice-Fragen ist jeweils nur **eine** richtige Antwort eindeutig anzukreuzen. Auf die richtige Antwort gibt es die angegebene Punktzahl.

Wollen Sie eine Multiple-Choice-Antwort korrigieren, kreisen sie bitte die falsche Antwort ein und kreuzen die richtige an.

Lesen Sie die Frage genau, bevor Sie antworten.

a) Was versteht man unter virtuellem Speicher?

2 Punkte

- Virtueller Speicher kann größer sein als der physikalisch vorhandene Arbeitsspeicher. Gerade nicht benötigte Speicherbereiche können auf Hintergrundspeicher ausgelagert werden.
- Virtueller Speicher wird vom Compiler beim Binden angelegt.
- Virtueller Speicher sind die nicht vorhandenen Bereiche des physikalischen Adressraums.
- Virtueller Speicher kann dynamisch zur Laufzeit von einem Programm erzeugt werden (Funktion `valloc(3)`).

b) Welche Aussage zu Semaphoren ist richtig?

3 Punkte

- Semaphore werden benutzt um in kritischen Abschnitten Interrupts zu sperren und so den gleichzeitigen Zugriff auf gemeinsame Datenstrukturen zu verhindern.
- P- und V-Operationen werden am besten eingesetzt, um Nebenläufigkeit zwischen Signalbehandlungsfunktionen und dem eigentlichen Programmablauf zu synchronisieren.
- Eine P-Operation kann in einem Anwendungsprogramm unter UNIX durch normale C-Anweisungen nicht implementiert werden.
- Die V-Operation dekrementiert den Semaphor um 1 und deblockiert andere in einer V-Operation blockierte Prozesse.

c) Man unterscheidet zwischen privilegierten und nicht-privilegierten Maschinenbefehlen. Welche Aussage ist **richtig**?

2 Punkte

- Privilegierte Maschinenbefehle dürfen in Anwendungsprogrammen grundsätzlich nicht verwendet werden.
- Die Benutzung eines privilegierten Maschinenbefehls in einem Anwendungsprogramm führt zu einer asynchronen Programmunterbrechung.
- Privilegierte Maschinenbefehle können durch Betriebssystemprogramme implementiert werden.
- Mit nicht-privilegierten Befehlen ist der Zugriff auf Geräteregister grundsätzlich nicht möglich.

d) Welche Aussage zu *Symbolic Links* ist richtig?

2 Punkte

- Symbolic Links sind nicht über Dateisystemgrenzen hinweg gültig.
- Wenn eine Datei gelöscht wird, werden auch alle auf sie verweisenden Symbolic Links automatisch mit aufgeräumt.
- Beim Zugriff auf einen Symbolic Link kann der Fehler "No such file or directory" auftreten obwohl der Link existiert.
- Die Anzahl der Symbolic Links, die auf eine Datei verweisen wird in deren Inode gespeichert.

e) Welche Aussage über den Rückgabewert von `fork()` ist richtig?

1 Punkt

- Der Sohn-Prozess bekommt die Prozess-ID des Vater-Prozesses.
- Im Fehlerfall wird im Sohn-Prozess -1 zurückgeliefert.
- Dem Vater-Prozess wird die Prozess-ID des Sohn-Prozesses zurückgeliefert.
- Bei erfolgreicher Ausführung kehrt `fork()` nicht zum Vater-Prozess zurück.

**Aufgabe 2: (15 Punkte)**

*Sie dürfen diese Seite und die Manual-Seite am Ende der Klausur zur besseren Übersicht bei der Programmierung heraustrennen!*

Erstellen Sie ein Programm **filesize**, das für alle übergebenen Dateien deren Größe (in Bytes) ausgibt. Sollte die Größe für eine Datei nicht ermittelt werden können, so wird hierfür eine sinnvolle Fehlermeldung ausgegeben und dann fortgefahren.

Das Ermitteln der Größe für eine einzelne Datei soll in einer eigenen Funktion `int getsize(const char *filename);` programmiert werden. Diese Funktion liefert als Ergebnis die Dateigröße. Im Fehlerfall wird `-1` zurückgegeben und die Fehlerursache ist in der globalen Variablen `errno` hinterlegt.

Beispiel:

Aufruf:

```
filesize datei1 daatei2 ../dir3/datei1 /tmp/filex
```

liefert

```
datei1: 348
daatei2: 7791
../dir3/datei1: Permission denied
/tmp/filex: 230436
```

```
/* includes */
#include <stdio.h>
#include <sys/types.h>
#include <sys/stat.h>
#include <unistd.h>
```

.....  
.....

```
/* Funktion main */
```

..... m

..... m

..... m

.....

.....

..... m

..... m

.....

.....

.....

.....

.....

.....

..... g

.....

..... g

.....

.....

..... g

.....

.....

.....

.....

.....

**Aufgabe 3: (5 Punkte)**

Wenn das folgende Programmstück in einem UNIX-System abläuft, wird ein Fehler auftreten.

```
...  
int *p = NULL;  
...  
*p = -1;  
...
```

Bitte möglichst knappe, aber präzise Antworten:

- a) Welcher Fehler tritt auf?  
.....  
.....
- b) Warum tritt der Fehler auf?  
.....  
.....  
.....
- c) Der Fehler wird von einer Hardwarekomponente zuerst entdeckt - welche Komponente ist das?  
.....  
.....
- d) Mit welchem Mechanismus wird der Fehler dem Betriebssystem mitgeteilt?  
.....  
.....
- e) Was macht das Betriebssystem mit dem Prozess, der gerade das Programmstück ausführt?  
.....  
.....